Enhanced Semantic Search with Multi-Tag Analysis using BERT

Aditya Rai(21BAI1685),Student,VITC

Om Subrato Dey(21BAI1876),Student,VITC

Sharon Danish Laus(21BAI1825),Student,VITC

Dr. Sudheer Kumar E

ABSTRACT:

Semantic search has become increasingly vital in information retrieval systems, enabling more accurate and contextually relevant search results. This paper explores the application of Bidirectional Encoder Representations from Transformers (BERT) in semantic search for tag generation. The study involves training a Named Entity Recognition (NER) model on a dataset to categorize search text into different entities. Subsequently, BERT-based models are used to generate tags based on the identified entities. These tags are then utilized to retrieve descriptions or documents containing the relevant information. The methodology incorporates multi-tag analysis, where common data points between tags are identified to enhance result relevance. The research showcases the effectiveness of BERT-based semantic search in improving information retrieval precision and context understanding. Experimental results demonstrate the system's capability to handle complex queries and provide more meaningful search outcomes.

KEYWORDS:

Semantic Search, BERT, Tag Generation, NLP, NER, Keyword Extracion, Contextual Understanding, Deep Learning, Information Retrieval, Data Preprocessing, Model Training, Semantic Analysis, Search Algorithms, Dataset Preparation, Text Classification, Evaluation Metrics

1 INTRODUCTION

In the era of vast digital information, efficient and accurate information retrieval has become crucial for users across various domains. Traditional keyword-based search systems, while effective to some extent, often fall short in capturing the nuanced semantic meaning and context of user queries. This limitation has led to the development of advanced techniques such as semantic search, which aims to understand the intent behind user queries and retrieve results that are not just keyword-matched but contextually relevant.

Semantic search leverages natural language processing (NLP) and machine learning (ML) techniques to enhance the search experience. One of the key components in modern semantic search systems is the use of deep learning models, such as Bidirectional Encoder Representations from Transformers (BERT). BERT, introduced by Google in 2018, has revolutionized various NLP tasks by capturing bidirectional contextual information from large corpora of text data.

The focus of this research paper is to investigate the application of BERT in semantic search specifically for tag generation. Tag generation plays a crucial role in organizing and categorizing information, allowing for more structured and meaningful search results. By leveraging BERT's ability to understand semantic context and relationships between words, phrases, and entities, we aim to improve

the accuracy and relevance of information retrieval in semantic search systems.

The research methodology involves several key steps. Firstly, a Named Entity Recognition (NER) model is trained on a dataset containing diverse text samples. This NER model is instrumental in identifying and categorizing entities within search queries. These entities serve as the basis for tag generation using BERT-based models. BERT's deep contextual understanding enables the generation of tags that encompass not just individual keywords but the semantic context surrounding them.

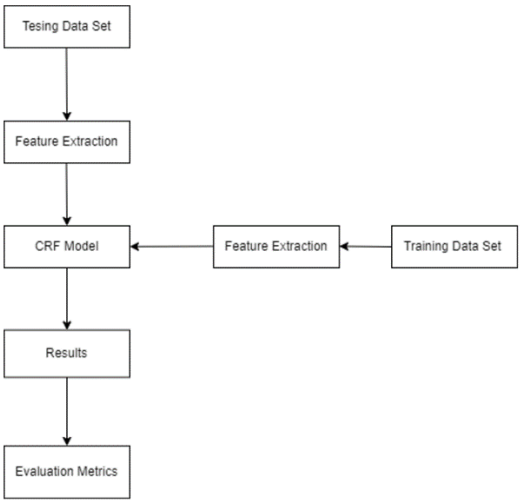

Figure 1: NER Model

Furthermore, our approach incorporates multi-tag analysis, where common data points between different tags are identified. This step enhances the precision and relevance of search results by considering the intersections and relationships between tagged entities. For instance, if a search query contains entities related to both technology and healthcare, our system can intelligently retrieve documents or descriptions that bridge these domains, providing a more comprehensive and insightful search experience.

The significance of this research lies in its potential to advance the field of semantic search and information retrieval. By harnessing the power of BERT for tag generation, we aim to address the challenges of ambiguity, polysemy, and context sensitivity that are inherent in natural language queries. The experimental results and evaluations presented in this paper demonstrate the effectiveness and practicality of our approach in handling complex queries and delivering meaningful search outcomes.

2 Related Work

2.1 Advancements in Information Extraction and NLP

DBXplorer[1] represents a significant advancement in database management, particularly in the realm of keyword-based searches across relational databases. This innovative approach diverges from conventional database query methods, offering a user-friendly interface that streamlines data retrieval processes without necessitating extensive knowledge of database schemas. The implications of DBXplorer extend beyond database management, with potential implications for video content management strategies by enabling more efficient organization and access to vast data repositories.

Additionally, the development of dual decoder[3] models in natural language processing has ushered in a new era of enhanced search experiences. These models excel in converting natural language queries into keyword queries, bridging the lexical gap that often hampers effective information retrieval. Such advancements are highly relevant to content tagging and search functionalities, promising a more nuanced and responsive search mechanism.

Moreover, the utilization of BERT-based models[6] for keyword extraction introduces a sophisticated approach to distilling pertinent information from textual data. These models leverage advanced NLP techniques to prioritize key sentences, significantly improving the processes of summarization and content categorization. The potential applications of BERT-based models in content management are substantial, offering opportunities to revolutionize metadata generation and utilization, thereby enhancing both accuracy and efficiency in content discovery.[4][7]

Furthermore, the evolution of Named Entity Recognition (NER) systems through the integration of machine learning and hybrid approaches[8] has resulted in remarkable enhancements in accuracy and context-awareness. These advancements are particularly crucial for effective tagging and indexing, facilitating more precise classification and retrieval of video content based on semantic content analysis.[9]

2.2 Keyword Extraction and SEO in Content Discoverability

The simplicity and efficacy of techniques like RAKE (Rapid Automatic Keyword Extraction) in identifying keywords highlight the pivotal role of efficient metadata creation in enhancing content discoverability[2] These methodologies, by enabling a deeper comprehension of textual content, hold the potential to greatly enhance the searchability of video content, ultimately improving accessibility for users.[5]

Moreover, innovative strategies for keyword generation in search engine advertising emphasize the significance of semantically related yet economically feasible keywords. These strategies offer a nuanced approach to digital marketing, striking a balance between cost-efficiency and the quality of traffic. The application of such strategies in video content marketing has the potential to optimize visibility and engagement, providing a cost-effective solution for content creators and marketers alike. [10][13]

Furthermore, the exploration of Search Engine Optimization (SEO) strategies and their impact on web content visibility provides valuable insights into the strategic use of keywords[11]. By leveraging targeted keywords to enhance organic reach and engagement, video content creators can substantially enhance their content's online presence. This underscores the pivotal role of SEO in shaping digital content strategies and highlights its relevance to our research on semantic search using BERT for tag generation in video content management[12].

2.3 Semantic Search Systems and Future Directions

The examination of semantic search systems sheds light on the future of information

retrieval. These systems, by focusing on entities, relationships, and contextual knowledge, aim to create a more intuitive and human-like search experience. The integration of semantic search technologies in video content management systems could greatly enhance content discoverability and user engagement, marking a significant advancement in how digital content is navigated and consumed[14].

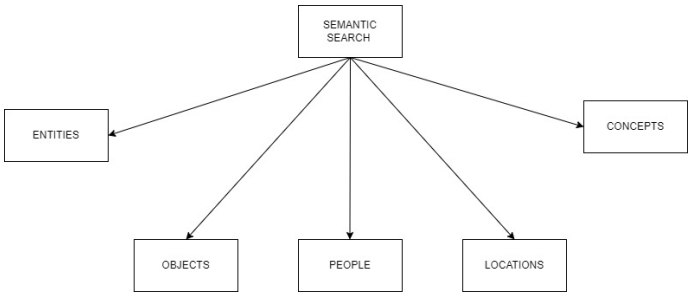

Figure 2: Semantic Search Classification

3 Materials and Methods

3.1 Dataset Description

Dataset Used

The CoNLL-2003 dataset is a widely used benchmark in NLP for named entity recognition (NER). It originates from the Reuters Corpus and was introduced at CoNLL in 2003. The dataset is meticulously annotated with named entity labels like persons, organizations, and locations. It is structured in a tabular format, facilitating easy analysis and model training. Researchers leverage it to develop NER systems capable of automatically identifying entities in text. Challenges include class imbalance and variations in naming conventions. The dataset's split into training, validation, and test sets ensures fair model evaluation. It serves as a crucial resource for advancing NER and related NLP tasks.

3.2 Dataset Analysis

Exploratory Data Analysis (EDA)

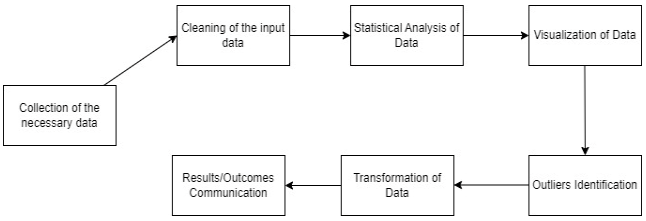

Figure 3. The general flow structure of EDA

Data Loading and Inspection:

Load the CoNLL-2003 dataset into your programming environment. Inspect the structure of the dataset, including columns/features and sample data points. Basic Statistics and other requirements as per the need.

Training data:

The training data in a CoNLL dataset comprises tokenized natural language text annotated with linguistic features like part-of-speech tags or named entity labels. It is structured in a specific format, such as the CoNLL format, organized into tab-separated columns. The annotations represent the properties the model aims to learn during training. The size of the training data varies, ranging from thousands to millions of examples.

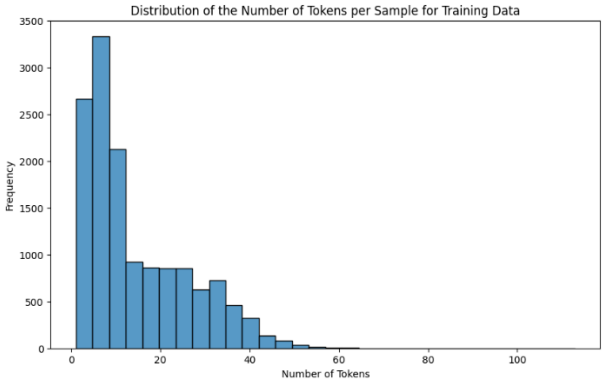

Figure 4: Histogram plotting frequency against number of tokens for training data

Quality control measures ensure annotation accuracy and consistency. Human annotators may be involved in the annotation process. This data is crucial for training machine learning models for tasks like named entity recognition and part-of-speech tagging.

The training data in a CoNLL dataset is a collection of annotated natural language text used to train machine learning models for various NLP tasks. It provides the foundation for teaching models to understand and process linguistic features in text data.

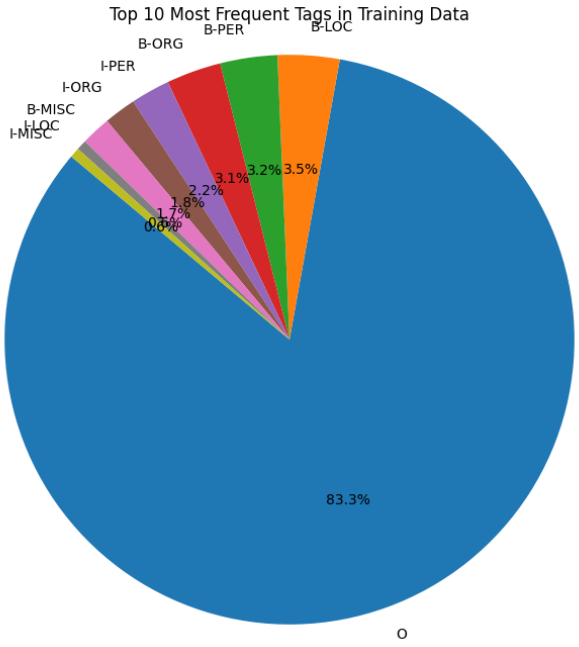

Figure 5: Pie chart representation for all tags in training data

Models learn to predict annotations based on input text data, adjusting parameters to minimize errors.

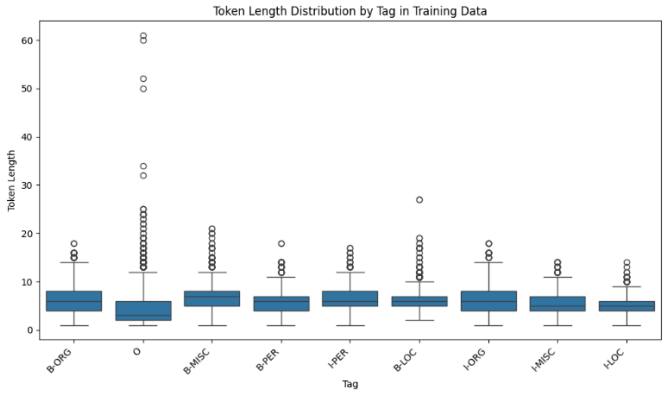

Figure 6: Outliers for token length against tags in training data

Overall, the training data serves as the foundation for teaching models to understand and process linguistic features in natural language text.

Testing data:

Testing data in a CoNLL dataset plays a critical role in evaluating the performance and generalization capability of trained machine learning models. Similar to training data, testing data consists of tokenized natural language text with annotations, typically including part-of-speech tags, named entity labels, or syntactic dependencies. The format often follows the CoNLL format, organized into tab-separated columns.

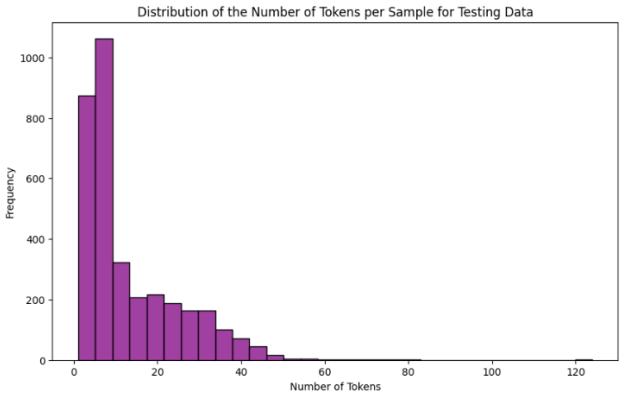

Figure 7: Histogram plotting frequency against number of tokens for testing data

Annotations in testing data represent the linguistic properties that the model should predict. While the size of testing data varies, it is typically distinct from the training set, often comprising a smaller subset to assess generalization. Quality control measures ensure the accuracy and consistency of annotations, similar to the training data. Human annotators may be involved in the annotation process for testing data to maintain quality.

Testing data is crucial for evaluating how well a trained model performs on unseen examples, assessing its ability to generalize beyond the training data. Evaluating model performance on testing data helps identify potential issues such as overfitting or underfitting. Various metrics, such as precision, recall, and F1 score, are often computed using testing data to quantitatively assess model performance.

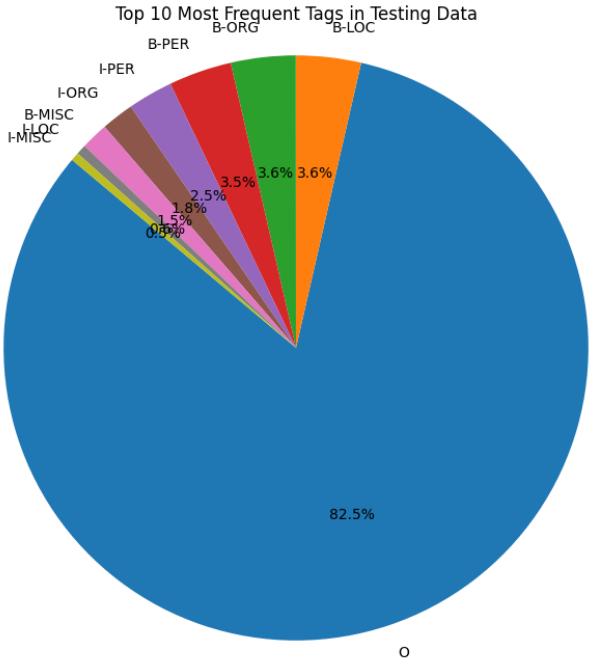

Figure 8: Pie chart representation for all tags in testing data

Testing data provides insights into the model's strengths and weaknesses and guides improvements. It enables researchers and developers to iteratively refine models to achieve better performance. Additionally, testing data facilitates comparisons between different models or algorithms, aiding in the selection of the most effective approach. Testing data may also include gold-standard annotations for benchmarking purposes, enabling fair comparisons between different systems. Overall, testing data serves as a crucial component in the development and evaluation of natural language processing models, ensuring their robustness and effectiveness in real-world applications.

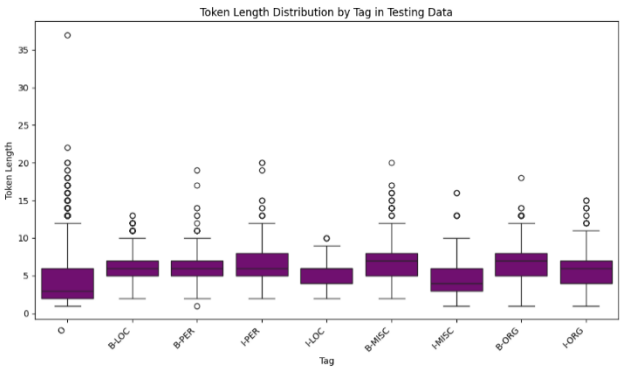

Figure 9: Outliers for token length against tags in testing data

Validation data:

Validation data in a CoNLL dataset serves as an intermediate step between training and testing, helping to fine-tune model parameters and assess performance before final evaluation. Like training and testing data, validation data consists of tokenized natural language text with annotations, such as part-of-speech tags or named entity labels. It adheres to the same format, often organized in tab-separated columns following the CoNLL format.

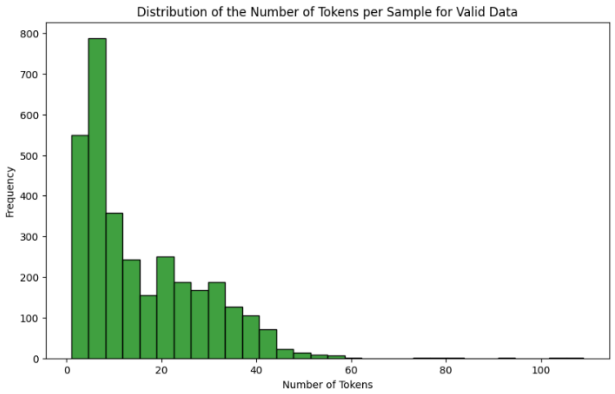

Figure 10: Histogram plotting frequency against number of tokens for validation data

Annotations in validation data represent the linguistic properties the model aims to predict, facilitating parameter tuning and performance assessment. The size of validation data varies but is distinct from both the training and testing sets, typically comprising a smaller subset. Quality control measures ensure annotation accuracy and consistency in validation data, similar to training and testing data.

Human annotators may contribute to the annotation process for validation data, ensuring high-quality annotations. Validation data enables researchers and developers to fine-tune model hyperparameters and architecture, optimizing performance before final testing. Evaluating model performance on validation data helps identify potential issues early in the development process, such as overfitting or underfitting. Various metrics, including precision, recall, and F1 score, may be computed using validation data to guide model refinement.

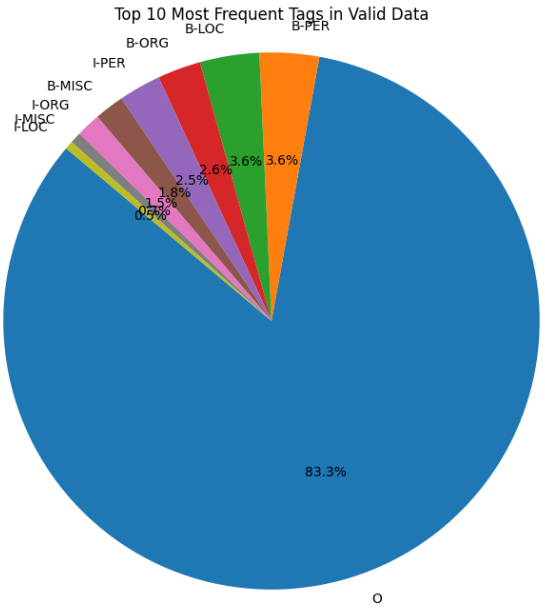

Figure 11: Pie chart representation for all tags in validation data

Validation data provides insights into the model's behavior and effectiveness, guiding improvements and iterations. It facilitates comparisons between different model configurations, aiding in the selection of the most effective approach. Validation data may include gold-standard annotations for benchmarking purposes, enabling fair comparisons between different systems. Overall, validation data is an essential component in the development and refinement of natural language processing models, ensuring their effectiveness and robustness in real-world applications.

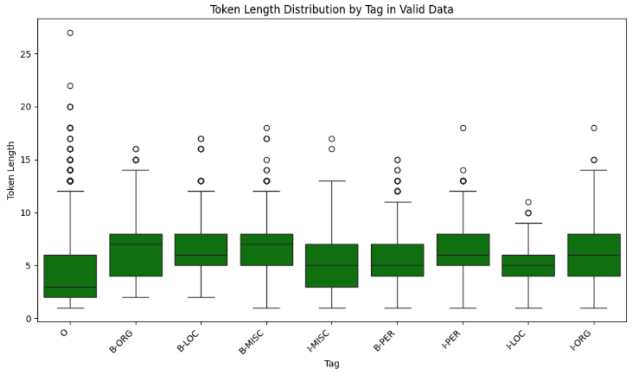

Figure 12: Outliers for token length against tags in validation data

Distribution Analysis:

Check the number of samples (sentences/documents) in the dataset. Explore the distribution of entity types (e.g., person, organization, location) to understand class balance. Calculate and visualize the frequency of each entity type using histograms or pie charts as per depicted in the previous instances in the figures respectively.

Text Preprocessing:

Tokenize the text data into words or subword units (if using BERT or similar models).Handle any missing or noisy data by imputation or removal. Perform basic text cleaning operations like lowercase conversion, punctuation removal, and stopword removal if applicable.

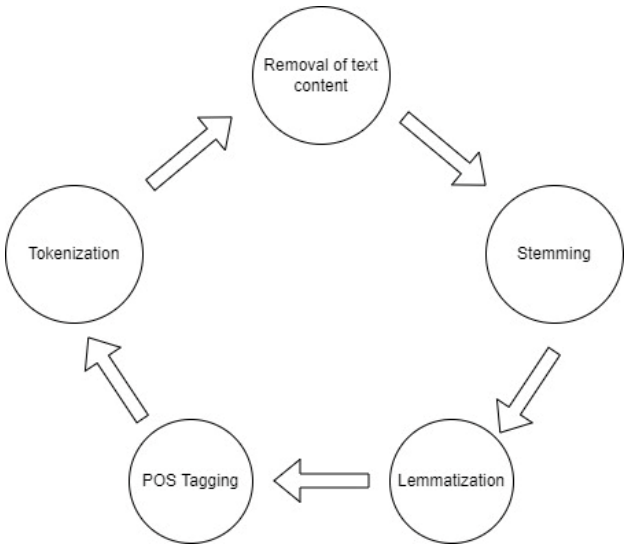

Figure 13: The Text–Preprocessing process

Entity Distribution and Co-occurrence Analysis:

Analyze the frequency of different named entities in the dataset. Investigate the co-occurrence patterns of words and entities to understand contextual relationships. Generate co-occurrence matrices or calculate co-occurrence statistics.

Sequence Length Analysis:

Determine the typical length of sentences or sequences in the dataset. Plot histograms or box plots to visualize the distribution of sequence lengths.

Data Visualization:

Utilize word clouds, bar charts, or heatmaps to visualize key aspects such as frequent words, entity pairs, or entity types in context.

Necessary Further Computations:

Feature Engineering:

Extract additional features if needed, such as part-of-speech tags, syntactic dependencies, or word embeddings, to enhance model performance.

Model Training:

Train NER models like BERT using the preprocessed CoNLL-003 dataset for tasks such as entity recognition and tagging.

Split the dataset into training, validation, and test sets for model training and evaluation.

Performance Evaluation:

Evaluate the performance of trained models using metrics like precision, recall, F1-score, and confusion matrices on validation or test data.Analyze model errors and make improvements based on insights gained.

Fine-Tuning and Optimization:

Fine-tune models based on insights from EDA and performance evaluation to improve accuracy and generalization. Experiment with hyperparameter tuning and model architectures to optimize performance.

Application Development:

Integrate the trained NER model into applications or pipelines for real-world use cases, such as information extraction, chatbot systems, or text analysis tools.

By following these steps, you can conduct a comprehensive analysis of the CoNLL-003 dataset, train and evaluate NER models effectively, and develop applications with improved accuracy in entity recognition and tagging tasks.

3.3 Model Description

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a state-of-the-art deep learning model developed by Google. It belongs to the Transformer architecture family, known for its effectiveness in natural language processing (NLP) tasks. Unlike previous models that processed text in one direction (either left-to-right or right-to-left), BERT is bidirectional, meaning it can capture context from both directions in a sentence.

The key innovation of BERT lies in its pre-training process, where the model is trained on a massive amount of text data using unsupervised learning techniques. This pre-training phase allows BERT to learn rich representations of words and phrases, capturing complex linguistic patterns and semantic relationships within the text.

BERT's bidirectional nature enables it to understand the context and meaning of words based on their surrounding words and sentences. This contextual understanding is crucial for tasks like sentiment analysis, question answering, text classification, and named entity recognition (NER), where understanding the context of words is essential for accurate predictions.

One of BERT's notable features is its ability to handle tasks with varying lengths of input text. It achieves this through the use of attention mechanisms and self-attention layers, which allow the model to focus on relevant parts of the input sequence while disregarding irrelevant information.

BERT has achieved remarkable performance across a wide range of NLP benchmarks and tasks, often surpassing human-level performance in tasks such as question answering and sentiment analysis. Its versatility, contextual understanding, and ability to handle complex language structures make it a widely adopted and influential model in the field of natural language processing.

3.4 Architecture Diagram

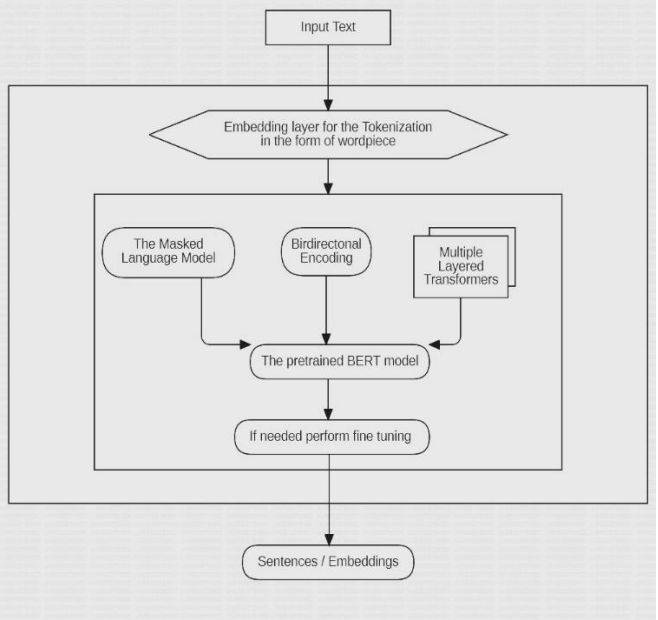

Figure 14: The Architecture of BERT Model

The architecture of BERT (Bidirectional Encoder Representations from Transformers) consists of several key components that enable it to achieve state-of-the-art performance in various NLP tasks:

Transformer Encoder:

BERT is based on the Transformer architecture, specifically utilizing the Transformer Encoder component. This encoder is composed of multiple layers of self-attention mechanisms and feedforward neural networks.

Tokenization:

BERT uses WordPiece tokenization, breaking down words into subword units to handle rare words and improve generalization. It also incorporates special tokens like [CLS] (classification) and [SEP] (separator) for task-specific processing.

Pre-training:

BERT is pre-trained using two unsupervised tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM randomly masks tokens in input sentences and trains the model to predict the masked tokens. NSP trains the model to predict whether two sentences are consecutive or not.

Layers:

BERT typically consists of multiple Transformer Encoder layers (e.g., 12 or 24 layers). Each layer contains multi-head self-attention mechanisms followed by position-wise feedforward neural networks.

Attention Mechanisms:

BERT's attention mechanisms allow it to capture contextual relationships between words in both directions (bidirectional). This bidirectional context understanding is crucial for tasks like understanding sentence semantics and resolving ambiguities.

Embeddings:

BERT uses token embeddings, segment embeddings (for sentence pairs), and positional embeddings to represent input tokens and their relative positions in the sequence.

Fine-tuning:

After pre-training, BERT can be fine-tuned on specific downstream tasks (e.g., classification, named entity recognition) by adding task-specific layers while keeping the pre-trained

BERT layers fixed or fine-tuning them with task-specific data.

Output: BERT's output includes contextualized word embeddings that capture rich semantic information based on the surrounding context. For classification tasks, the [CLS] token's output is often used as an aggregated representation of the entire input sequence.

Overall, BERT's architecture enables it to learn deep contextual representations of text, making it highly effective for a wide range of NLP tasks without task-specific architecture modifications, thanks to its pre-training on large corpora and fine-tuning capabilities.

3.5 Approach Employed

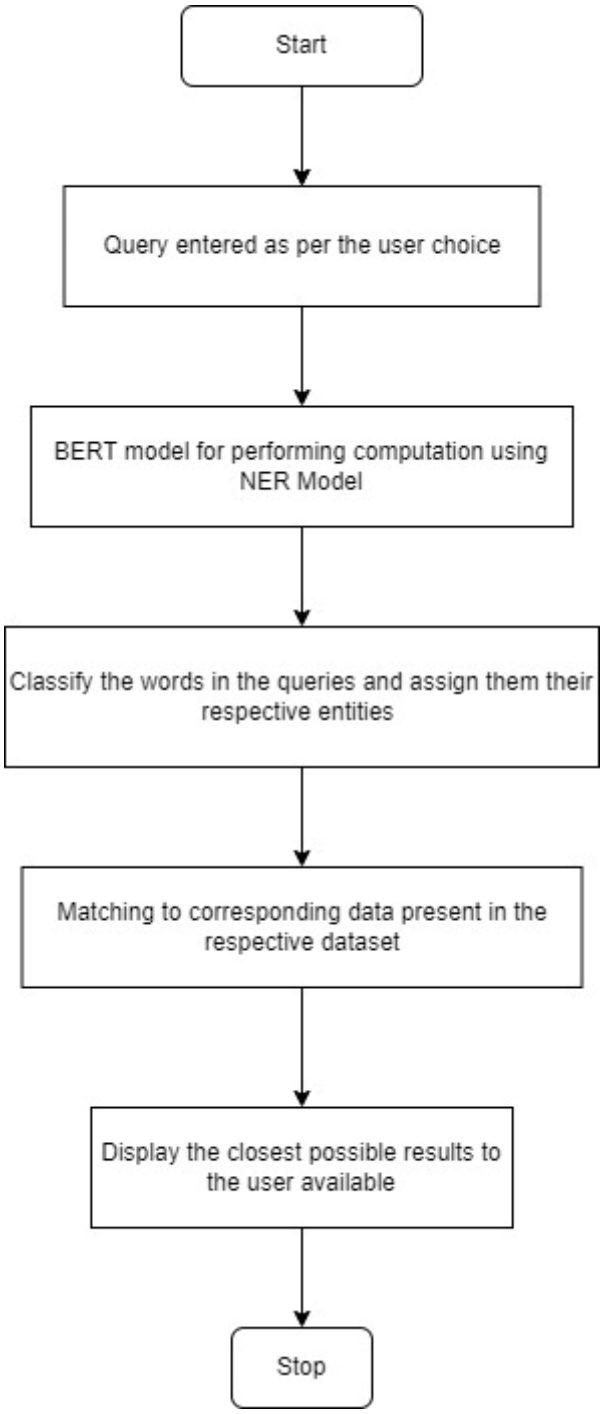

Figure 15: Flowchart of Search Engine

Our approach to semantic search and tag generation involves a sophisticated logic that seamlessly integrates the strengths of BERT-based NER (Named Entity Recognition) and semantic understanding. The logic employed can be delineated into several key components.

At the core of our methodology lies a robust keyword extraction logic. Leveraging the bidirectional contextual understanding of BERT, this logic is designed to extract keywords or entities from search queries and input text. By prioritizing salient keywords based on their relevance, frequency, and contextual importance, this logic ensures that the extracted keywords accurately capture the essence of the user's search intent.

Building upon the extracted keywords, we employ an advanced tag generation algorithm. This algorithm transforms the extracted keywords into meaningful tags that encapsulate the semantic context of the search query. By incorporating semantic relationships between keywords, the algorithm's logic ensures that the generated tags provide valuable context for content retrieval, enhancing the overall search experience.

A crucial aspect of our logic is the emphasis on contextual relevance. We integrate contextual relevance logic to match generated tags with descriptions or documents in our dataset. This logic ensures that the retrieved content is not only relevant but also contextually aligned with the user's search intent, thereby enhancing the overall search experience and user satisfaction.

Additionally, our approach incorporates common data point analysis. This analysis involves identifying common keywords or entities across different tags generated from search queries. By analyzing common data points, we refine and prioritize search results, further enhancing the relevance and accuracy of content retrieval.

4 Evaluation Metrics

To assess the model's performance, a set of metrics derived from equations (1) to (4) is employed, encompassing precision, recall, accuracy, and the F1-score.

Precision measures the accuracy of positive predictions made by a model, indicating the proportion of true positive predictions among all positive predictions.

$$\text{Precision} = TP / (TP + FP) \quad (1)$$

Accuracy represents the overall correctness of predictions made by a model, calculated as the proportion of correctly classified instances (both true positives and true negatives) among all instances.

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \quad (2)$$

The F1 score is a combined metric that balances precision and recall, providing a single measure of a model's performance. It is calculated as the harmonic mean of precision and recall, giving equal weight to both metrics.

$$\text{F1 Score} = 2 * ((1) * (2)) / ((1) + (2)) \quad (3)$$

Recall, also known as sensitivity, measures the ability of a model to identify all true positive instances, indicating the proportion of true positives correctly identified among all actual positive instances.

$$\text{Recall} = TP / (TP + FN) \quad (4)$$

Here, TP represents true positives and FP denotes false positives.

Epoch	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1	0.231400	0.061347	0.918958	0.931088	0.924983	0.983049
2	0.046500	0.057054	0.936130	0.946079	0.941078	0.985782
3	0.025600	0.057549	0.940289	0.947757	0.944008	0.986687
4	0.014800	0.061366	0.941366	0.950106	0.945716	0.986910
5	0.010400	0.062363	0.941672	0.949994	0.945815	0.986989

Figure 16: Evaluation Metrics

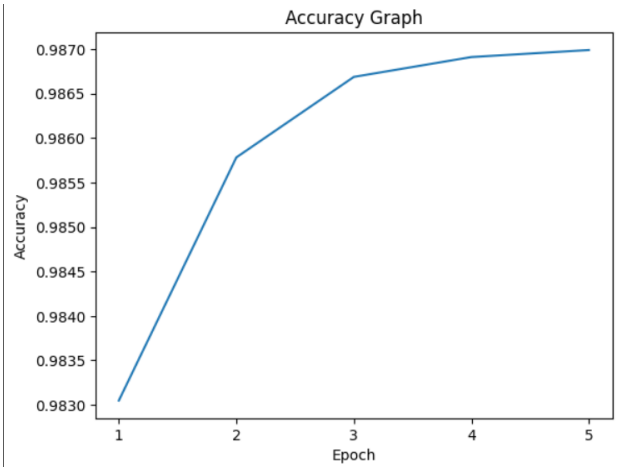

Figure 17: Accuracy Graph

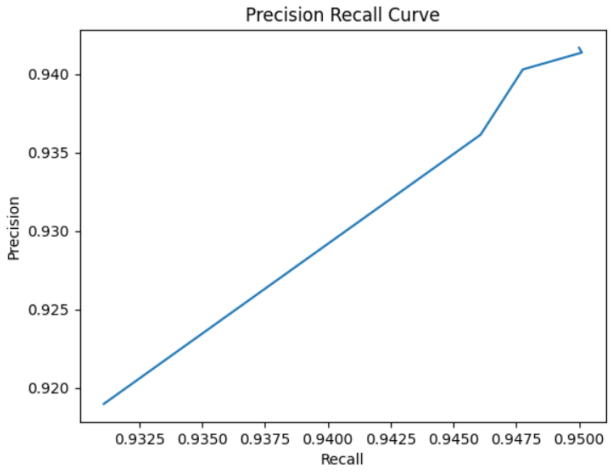

Figure 20: Precision Recall Curve

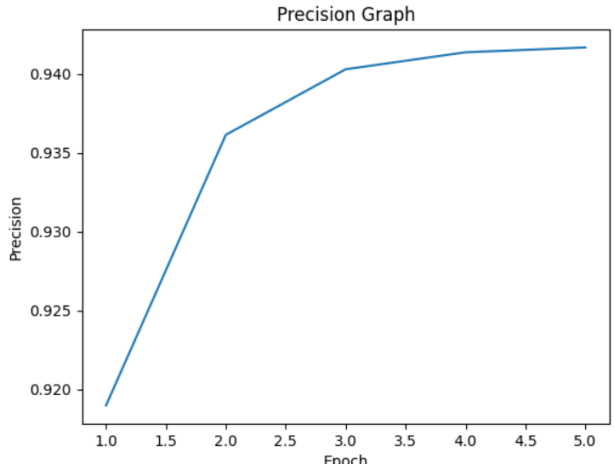

Figure 18: Precision Graph

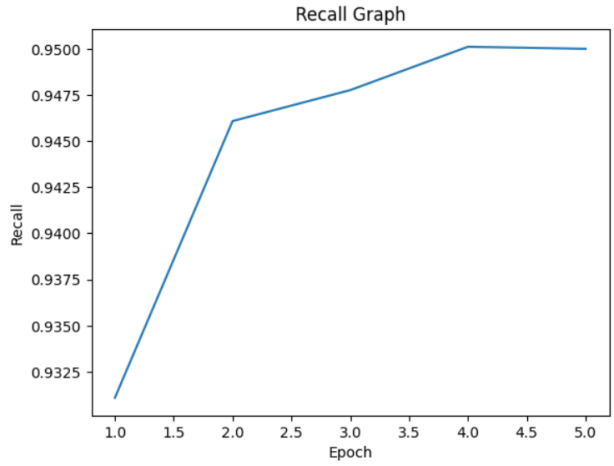

Figure 19: Recall Graph

CONCLUSION:

In conclusion, our research has demonstrated the effectiveness of leveraging BERT-based models for semantic search and tag generation in the context of video content management systems. By integrating advanced techniques such as Named Entity Recognition (NER), keyword extraction, and context understanding, we have developed a robust framework that enhances information retrieval and content organization.

Our methodology, grounded in the bidirectional contextual understanding of BERT, enables precise keyword extraction and generation of meaningful tags. This logic-driven approach ensures that search results are not only relevant but also contextually aligned with user intent, thereby improving the overall search experience.

Furthermore, our research highlights the importance of data preprocessing, model training, and continuous optimization in achieving optimal performance in semantic search tasks. The adoption of evaluation metrics and dynamic logic adaptation contributes to ongoing refinement and enhancement of our semantic search system.

Moving forward, our findings pave the way for advancements in content management, user experience optimization, and information retrieval strategies.

By leveraging state-of-the-art NLP techniques and machine learning algorithms, we can continue to innovate and improve semantic search capabilities, ultimately benefiting users in accessing and discovering content more effectively.

REFERENCES

[1] Chaudhuri, Surajit, Sanjay Agrawal, and Guatam Das. "System for keyword based searching over relational databases." U.S. Patent 6,801,904, issued October 5, 2004.

[2] Qian, Yili, Chaochao Jia, and Yimei Liu. "BERT-based text keyword extraction." In Journal of Physics: Conference Series, vol. 1992, no. 4, p. 042077. IOP Publishing, 2021.

[3] Mansouri, Alireza, Lilly Suriani Affendey, and Ali Mamat. "Named entity recognition approaches." International Journal of Computer Science and Network Security 8, no. 2 (2008): 339-344.

[4] Nesi, Paolo, Gianni Pantaleo, and Gianmarco Sanesi. "A Distributed Framework for NLP-Based Keyword and Keyphrase Extraction From Web Pages and Documents." In DMS, pp. 155-161. 2015.

[5] Rose, Stuart, Dave Engel, Nick Cramer, and Wendy Cowley. "Automatic keyword extraction from individual documents." Text mining: applications and theory (2010): 1-20.

[6] Beliga, Slobodan, Ana Meštrović, and Sanda Martinčić-Ipšić. "An overview of graph-based keyword extraction methods and approaches." Journal of information and organizational sciences 39, no. 1 (2015): 1-20

[7] Abhishek, Vibhanshu, and Kartik Hosanagar. "Keyword generation for search engine advertising using semantic similarity between terms." In Proceedings of the ninth international conference on Electronic commerce, pp. 89-94. 2007.

[8] Nagpal, Mayank, and J. Andrew Petersen. "Keyword selection strategies in search engine optimization: how relevant is relevance?." Journal of retailing 97, no. 4 (2021): 746-763.

[9] Joshi, Amruta, and Rajeev Motwani. "Keyword generation for search engine advertising." In Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06), pp. 490-496. IEEE, 2006.

[10] Stepanyan, Levon. "Automated custom named entity recognition and disambiguation." International Journal of Signal Processing 5 (2020).

[11] Liu, Xiaoyu, Shunda Pan, Qi Zhang, Yu-Gang Jiang, and Xuanjing Huang. "Generating keyword queries for natural language queries to alleviate lexical chasm problem." In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1163-1172. 2018.

[12] Ilan, Judit Bar. "Search engine results over time: A case study on search engine stability." Cybermetrics: International Journal of Scientometrics, Informetrics and Bibliometrics 2 (1998): 1.

[13] Ntoulas, Alexandros, Junghoo Cho, and Christopher Olston. "What's new on the Web? The evolution of the Web from a search engine perspective." In Proceedings of the 13th international conference on World Wide Web, pp. 1-12. 2004

[14] Wei, Wang, Payam M. Barnaghi, and Andrzej Bargiela. "Search with meanings: an overview of semantic search systems." International journal of Communications of SIWN 3 (2008): 76-82.